Ref. Ares(2020)2808211 - 29/05/2020

enabling new Demand REsponse Advanced, Market oriented and secure technologies, solutions and business models

DELIVERABLE: D3.7 Consumption flexibility models and aggregation techniques V2

Authors: Tudor Cioara (TUC), Ioan Salomie (TUC), Andreea Vesa (TUC)

Imprint

D3.7 Consumption flexibility models and aggregation techniques V2 (Month 29)

Contractual Date of Delivery to the EC:	31.05.2020
Actual Date of Delivery to the EC:	31.05.2020
Author(s):	Tudor Cioara (TUC), Ioan Salomie (TUC), Andreea Vesa (TUC)
Participant(s):	Contributors: Ionut Anghel (TUC), Marcel Antal (TUC), Claudia Pop (TUC), Viorica Chifu (TUC), Cristina Pop (TUC)
Project:	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM)
Work package:	WP3 – Techniques for DR Energy Flexibility Assessment
Task:	T3.3: Multi-energy Distributed generation Modelling and Virtual Power Plants
Confidentiality:	Public
Version:	0.99

Legal Disclaimer

The project enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 774478. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC). INEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

Table of Contents

List of Figures 4
List of Tables 5
List of Acronyms and Abbreviations 6
Executive Summary 7
1 Introduction 8
1.1 Purpose 8
1.2 Relation to other activities 8
1.3 Structure of the document 9
2 Dynamic VPP Coalitions 10
2.1 VPP model and fog-based architecture 10
2.2 Prototype Implementation 13
2.3 Evaluation results 22
3 Blockchain based VPP 25
3.1 Smart contracts for decentralized coalition formation 25
3.2 Prototype implementation and results 30
4 Conclusion 33
References 34

List of Figures

Figure 1. eDREAM pert diagram showing WP3 relations with other work packages and Task 3.3 8
Figure 2. VPP coalition optimization problem as a CSP 11
Figure 3. Fog computing arhitecture for VPP model optimization 12
Figure 4. VPP dynamic coalitions formation protype 13
Figure 5. Sequence diagram for VPP prototype modules interaction 14
Figure 6. Information of prosumers portofolio available for the VPP manager 16
Figure 7. VPP manager interface for selecting a specific service from the list provided.................. 17
Figure 8. Interface showing information about the VPP coalition of prosumers created to trade energy 18
Figure 9. Web page showing information about the VPP coalition of prosumers created for capacity selling 19
Figure 10. Web page showing information about grid power factor before and after VPP coalition deployment 20
Figure 11. Web page showing information about the VPP coalition of prosumers created delivery of reactive power control 21
Figure 12. Web page showing information about the VPP coalition of prosumers created for addressing a DR service... 22
Figure 13. Scenario considered for evaluation 22
Figure 14. Power factor improvement as result of VPP construction and service delivery 23
Figure 15. New created hierarchical VPPs: (LEFT) maximum and minimum energy consumption in the service interval and (RIGHT) detailed energy profiles 23
Figure 16. Micro-grid total active power and VPP contribution 24
Figure 17. Number of messages send during one hour by smart meters associated with prosumers 24
Figure 18. Interaction between smart contracts in a hierachical VPP structure 26
Figure 19. Sequence diagram for decentralized VPP coalition formation using smart contracts 27
Figure 20. Smart contract method for publish a service request offer 28
Figure 21. Smart contract method for receiving a join VPP coalition offer 28
Figure 22. Smart contract method for finishing the VPP hierarchy construction process 29
Figure 23. Smart contract method for energy and financial settlement of the delivery..................... 29
Figure 24. Blockchain based VPP architecture 30
Figure 25. Hierarchical VPP structure created to deliver energy 31
Figure 26. VPP6 root state and aggregated lower level VPPs 31
Figure 27. VPP5 state on level 1 and aggregated lower level VPPs/prosumers 32
Figure 28. VPP3 state on level 1 and aggregated lower level VPPs/prosumers 32

List of Tables

Table 1. Modelled target services objectives to be met by constructed VPP 10
Table 2. VPP prototype requiments 16
Table 3. Experimental setup for prosumers generation 23

List of Acronyms and Abbreviations

aFRR	Automatic Frequency Restoration Process
API	Application Programming Interface
CHP	Combined heat and power
CSP	Constraint Satisfaction Problem
DEG	Distributed Energy Generators
DER	Distributed Energy Resources
DoD	Depth of Discharge
DR	Demand Response
DSO	Distributed System Operator
eDREAM	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models
ESS	Energy Storage System
FDA	Flexible Energy Demand Assets
IoT	Internet of Things
NP	Non-deterministic polynomial-time
PF	Power Factor
RES	Renewable Energy Sources
REST	Representational State Transfer
UPS	Uninterruptible Power Supply
VPP	Virtual Power Plant
WP	Work Package

Executive Summary

In this deliverable, we present the work done in the direction of implementing models and techniques for the aggregation of energy prosumers in VPPs, by constructing dynamic coalitions to address and deliver specific energy services.

The work follows up and builds upon the results reported in the first techniques version (deliverable D3.3) in which we had defined a new VPP model which allows the formalization of dynamic construction of coalitions of prosumers in VPPs as a constraint satisfaction problem which can be tailored and adapted to different types services to be offered and a hybrid optimization technique which combines the gradient-based solutions with nature-inspired heuristics for solving the optimization problem.

Specifically, we have implemented and integrated the VPP creation model and optimization in a software prototype that allows the construction of VPP coalitions for delivering the following services: trade energy, capacity bidding / selling, reactive power control, and VPP demand response. To address the on-the-fly construction of a VPP which combines and coordinates various types of prosumers to deliver services with closer real-time constraints, we have defined a fog-based architecture for deploying purposes. It features a hierarchy of three layers: the prosumers energy monitoring layer, the fog layer where the software prototype is executed and VPPs are constructed considering both the prosumers' local constraints and service level constraints and finally the cloud layer where the energy services are defined. We have evaluated our solution capability to solve at the fog level, the VPP specific constraints satisfaction problems, and by generating the prosumers coalitions at different levels in the hierarchy for delivering reactive power control services. The results show that the provided solutions can construct VPPs coalitions with low computational overhead (i.e. number of messages exchanged).

At the same time, we have defined a decentralized blockchain-based solution for the construction of VPP coalitions of prosumers. We have addressed the VPP coalition construction using a lightweight mechanism implemented leveraging on smart contracts that target the construction of hierarchical VPP structures. The hierarchical VPP has a tree-like structure, with a root VPP which is the initiator of the coalition and lower scale VPPs and all intermediary levels while the prosumers will be on the leaf. The smart contracts are also managing the service delivery and settlement process by tracking the actual levels of energy provided and by implementing a mechanism for increasing or decreasing the risk and reward/penalty, while at the same time considering the uncertainty of the energy forecasting process. The solution was implemented in a prototype and integrated with the eDREAM blockchain platform the evaluation results showing that it can be used for the delivery of capacity services. The advantage of this approach is that it is lightweight and does not need high computation time, so it is feasible to run on the blockchain, without third-party software outside the blockchain.

1 Introduction

1.1 Purpose

This deliverable provides an overview of the work done for defining models and optimization techniques aiming to create coalitions of prosumers in VPPs targeting to deliver energy services. The report summarizes the activities carried out after the release of D3.3 [5] in the direction of implementing and integrating the VPP model and optimization heuristics into a software prototype which targets different types of energy services. Other significant contribution reported is the fog-based solution which allows the construction of the VPP as close as possible to the prosumers edge level and a blockchain based solution for on-the-fly VPP hierarchical coalitions of prosumers construction leveraging on self-enforcing smart contracts.

The work has been done in relation with eDREAM project Task 3.3 “Multi-energy Distributed Generation Modelling and Virtual Power Plants” part of Work Package 3 “Techniques for DR and Energy Flexibility Assessment”.

1.2 Relation to other activities

WP3 uses the outputs of WP2 in terms of requirements and use-cases and implements the models and techniques that will provide the underlying base for developing the eDREAM envisioned next generation of demand response services both in a classical, centralized approach (WP4) but also in an innovative decentralized blockchain based manner (WP5) as depicted in Figure 1**Error! Reference source not found.**.

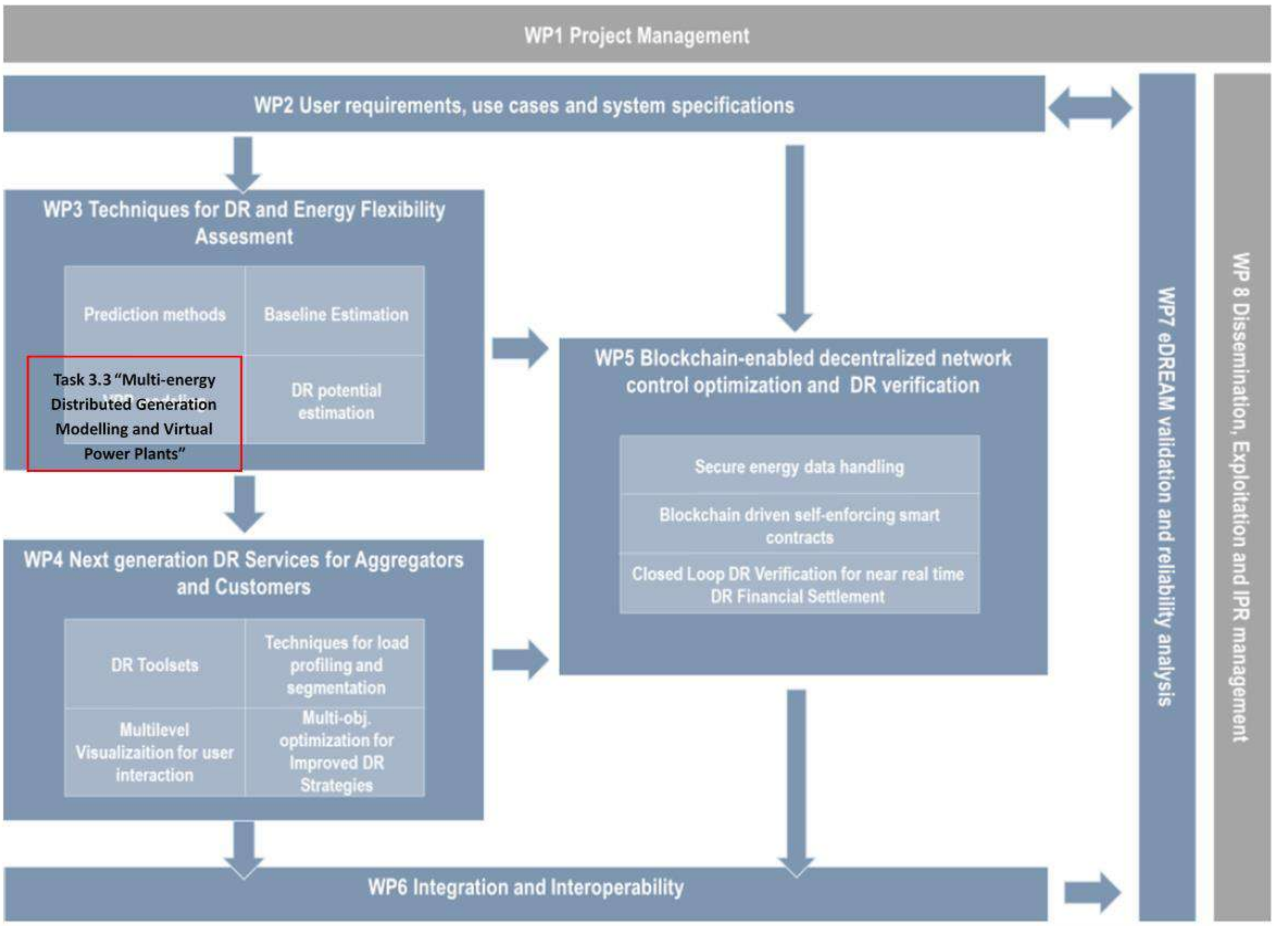


Figure 1. eDREAM pert diagram showing WP3 relations with other work packages and Task 3.3

In particular, the VPP model and decentralized optimization techniques will consider the outputs of the energy production/consumption forecasting tool developed in Task 3.1 that are implemented as a standalone component or integrated within the blockchain-based platform developed in WP5 benefiting on the advantages brought by the use of smart contracts.

1.3 Structure of the document

The remainder of the report is organized as follows:

- Section 2 presents an overview of the VPP model and optimization heuristic and a fog enabled architecture allowing the decentralized construction of a VPP hierarchy. Also, prototype implementation and user interaction guidelines are provided as well as evaluation results for the fog enabled deployment;
- Section 3 shows a blockchain-based approach for the decentralized construction of VPP coalitions of services and results for addressing a capacity selling service;
- Section 4 concludes the deliverable.

2 Dynamic VPP Coalitions

In this section we present the VPP creation and optimization model implementation in a prototype which allows the construction of VPP coalitions for addressing different energy services constraints. We start by presenting a short overview of the VPP model introduced in D3.3 and a fog-based architecture for decentralizing the VPP coalitions construction process towards the edge (see Section 2.1) and continue with the protype implementation details (see Section 2.2) and results considering the constraints of a frequency regulation service (see Section 2.3).

2.1 VPP model and fog-based architecture

In this sub-section we will summarize the VPP construction and optimization model presented in detail in D3.3 [5] and we will introduce a fog-based architecture allowing the optimization problem decentralization and solving towards the edge.

The VPP model considers three types of distributed energy prosumers that may be potentially coalized and for each of them we have modelled their local specific constraints.

For the **Distributed Energy Generators (DEG)** such as small-scale wind power plants, photovoltaic units, CHP systems, diesel generators, etc. the model considers the following parameters for specifying the local constraints of asset operation: (i) the forecasted energy generation values, E_K; (ii) the lower and upper levels of uncertainty considered in the forecasting process, U_L and U_H (iii) the maximum energy generation $E_{MAX}^{generation}$ and (iv) the cost of energy generation, Gen_{Cost}.

For the **Energy Storage Sources (ESS),** such as such as batteries and UPS, the model considers the following parameters for specifying the local constraints of asset operation: (i) maximum capacity: $MAX_K^{Load}[kWh], k \in \{1..S\}$, (ii) Depth of Discharge: $DoD_k, k \in \{1..S\}$, (iii) initial capacity and actual state: $ESS_k^{init}[kWh], ESS_k[kWh],\ k \in \{1..S\}$, (iv) maximum charge and discharge rates on time interval: $MAX_K^{Charge}[kWh], MAX_K^{Discharge}[kWh],\ k \in \{1..S\}$, (v) actual charging and discharging rates on a time interval: $C_{ESS}^k[kWh], D_{ESS}^k[kWh],\ k \in \{1..S\}$, (vi) charge and discharge loss factors: $\varphi_C, \varphi_D \in [0,1]$ and (vii) charge and discharge cost per unit: $COST_k^D\left[\frac{€}{[kWh]}\right], COST_k^C[\frac{€}{[kWh]}], k \in \{1..S\}$.

For the **Flexible Energy Demand Assets**, the model considers the following parameters for specifying the local constraints of asset operation: the baseline energy consumption of the flexible asset $E_K^{baseline}$; the lower and upper bounds of flexibility availability defined as energy values measured below or above the baseline $APC_{Below}^{flexibility}$ and $APC_{Above}^{flexibility}$ and the maximum energy consumption of the flexible asset $E_{MAX}^{flexibility}$.

The goal of the optimal coalition of prosumers construction process is to select a subset of the energy prosumers from local grid available portfolio which fulfils best the objectives defined for the type of energy service that is targeted to be delivered by the VPP (see Table 1), while meeting each energy prosumer local defined constraints.

Table 1. Modelled target services objectives to be met by constructed VPP (more details can be found in D3.3)

Target Service	Description	VPP Objective Function
Trade Energy	VPP is created to sell extra energy production on the market when the prices are high or to buy	$VPP_{profit}^{trading}(t) = \sum_{t=1}^{T} E_{VPP}^{generation}(t) * price(t) + R_{ESS}(C_{ESS}, D_{ESS}, price) - Gen_{Cost}$

	energy when the prices are low and store it in ESS.	
Capacity Bidding	In case a power plant cannot meet its commitment and needs to purchase replacement capacity a dynamically created VPP may offer it.	$min(\sqrt{\sum_{t=1}^{T}(E_{VPP}(t) - Target_{capacity})^2})$ $\max VPP_{profit}^{capacity} = Compensation * \sum_{t=1}^{T} E_{VPP}(t) - Gen_{Cost}$
Reactive power compensation	Frequency regulation. Unused capacity which can be activated to modify the reactive power. DEGs of specific type can be used to offer frequency balancing by injecting inductive reactive power in the grid.	$\min\left(\sqrt{\sum_{t=1}^{T}\left(PF_{target}(t) - PF_{grid}(t)\right)^2}\right)$ $\max VPP_{profit}^{PF} = Service_{Reward} - Gen_{cost}$
VPP demand response	Group energy prosumers to offer a demanded energy supply amount over a time window also considering the potential flexibility of FDAs.	$\min \sqrt{\sum_{t=1}^{T}\left((E_{VPP}^{generation}(t) + E_{VPP}^{flexibility}) - DSO_{Demand}(t)\right)^2}$

The optimization problem is NP-Complete, with the search space being 2^N (i.e. the set of all subsets that can be formed with elements of an initial set of N prosumers). Moreover, the VPP coalition formation for specific services is modelled as a CSP (see Figure 2) thus to efficiently solve it a hybrid gradient and population-based heuristic technique has been defined.

We aim at minimizing an objective function f having n real arguments and m integer arguments while fulfilling a set of k constraints of the form $c_i(x, y) \leq d$, knowing that each real and integer variable argument is bounded by a set of lower and upper limits.

Determine $x \in R^n, y \in Z^m$

$minimize\ VPP\ objective\ function\ (f(x,y)), f: R^n \times Z^m$

Such that the model defined constraints are meet

Service and Local Asset Constraints: $c_i(x, y) \leq d, i \in \{1..K\}, d \in R$

Variable Operation Parameters Bounds: $x_L \leq x \leq x_H,\ y_L \leq y \leq y_H$

Variable Types: $x \in X \subseteq R^n,\ \ y \in Y \subseteq Z^m$

Figure 2. VPP coalition optimization problem as a CSP

The proposed hybrid solution (see D3.3 for more details) uses a heuristic method/algorithm to compute the integer variable values, it fixes them as constants, making the f function differentiable, thus gradient-based methods are adopted being suitable to compute an approximate solution. The advantage of this approach is given by the potential to parallelize and decentralize the optimization problem computations to enhance the performance and decrease the time overhead. In case of a network of nodes capable of performing the computations (i.e. processing nodes located at each prosumer), the population of the algorithm can be split between the nodes, each of them running the gradient-based algorithm and then evolving its local individuals. At each round, the nodes communicate to exchange their local population, and each of them performs a ranking and a selection of the best individuals, generates a new population and distributes the best individuals to be ranked by the other nodes. Then, the population is split again between the processing nodes, and the gradient search is performed locally to increase the computation speed.

To address the on-the-fly construction of VPP which usually combines and coordinates local energy production sources with energy storage systems and flexible assets featuring controllable loads to deliver services with near real time constraints, we have defined a fog-based architecture for deploying the VPP model. The proposed architecture is based on a hierarchy of three layers (see Figure 3):

- An edge layer contains the physical IoT energy metering devices associated with each individual prosumer (i.e. energy generation, storage or flexible asset) from the smart grid. The local operational constraints of each individual asset are provided to the fog layer devices.
- A fog layer contains computational resources associated with a local geographical area (i.e. microgrid area) and the VPP construction and optimization model and heuristics deployed enabling the construction of virtual coalitions of prosumers in VPPs to provide various services for the main grid;
- A cloud layer represented by a cloud server on which the DSO runs its analytics to define specific services for optimal management of grid resources which can be addressed by a virtual coalition of prosumers. The objectives associated with each individual service to be addressed are provided to the fog layer devices.

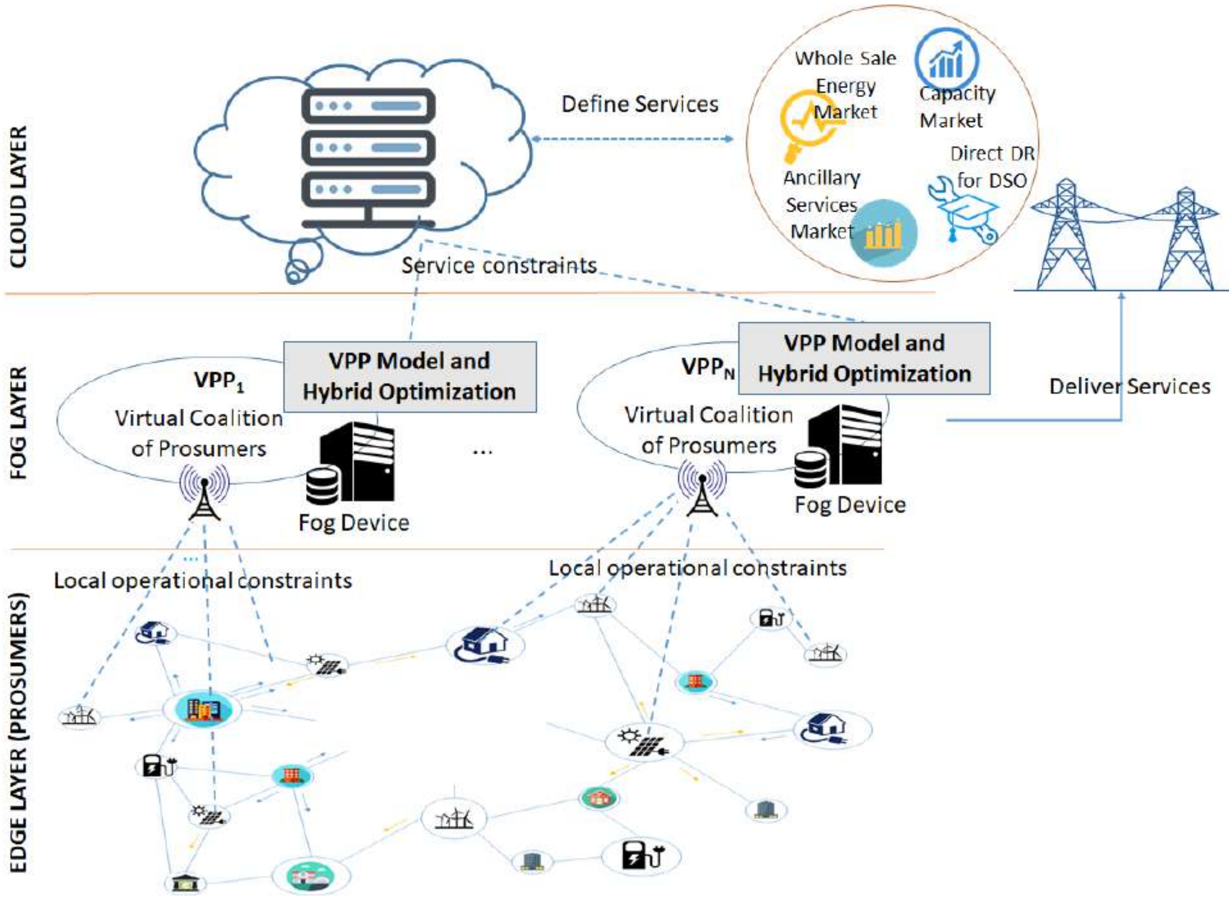


Figure 3. Fog computing arhitecture for VPP model optimization

This approach brings the following advantages:

- Proximity to the prosumers – the analytics for solving the optimization problems are run closer to the prosumers thus minimizing the decision time and providing the opportunity for selecting prosumers to participate in VPP coalitions no matter of their size or scale;

- Increased Locality – the virtual coalitions can be associated with a local microgrid and, as a result, addressing locally the potential management problems and avoiding their escalation to grid higher levels;
- Reduced Latency – in the data traffic from the lower edge level to the higher fog level (where the VPPs are constructed) and vice-versa. This is also an important condition of the availability of a high amount of data from the smart meters.

2.2 Prototype Implementation

In this section, we will provide details on the implementation of the software prototype dealing with the optimal construction of coalitions of prosumers in VPPs for meeting specific energy services constraints. Figure 4 below shows the final design of the prototype and the main technologies used for implementation.

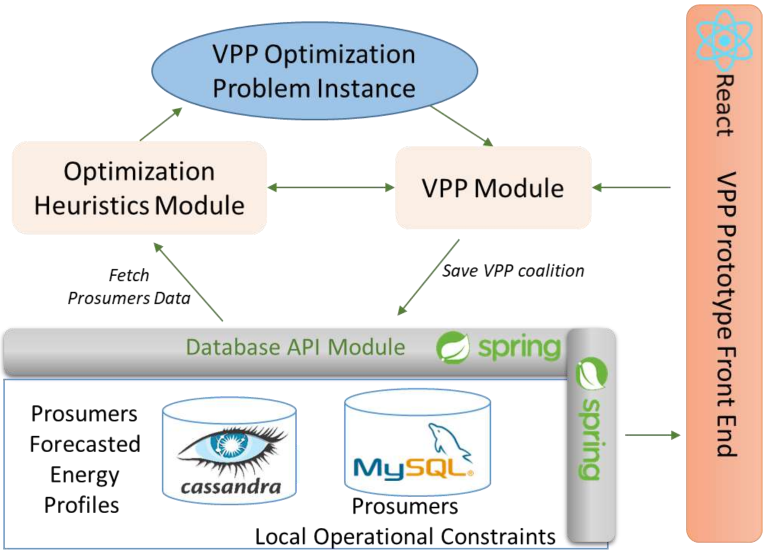


Figure 4. VPP dynamic coalitions formation protype

Database API Module has the role of providing a unified REST API for eDREAM distributed database which will store the prosumers monitored and forecasted energy profiles and their local operational constraints / parameters.

VPP Module receives the VPP coalition creation request, extracts the service level constraints as well as the prosumers local ones and formalizes the creation of optimal coalitions of prosumer as a CSP using the model we have described in Section 2.1.

Optimization Heuristics Module contains a library of heuristics used for solving the CSP problems for constructing optimal coalitions of prosumers in VPPs to deliver a service.

VPP Optimization Problem Instance defines the CSP problem that needs to be solved and implements all the data models needed by our hybrid gradient enhanced heuristics optimization technique, such as the binary expression tree and mathematical operations needed in optimization such as functions derivation.

VPP Prototype Front End is a module that is used to provide a web-based graphical user interface for the VPP manager.

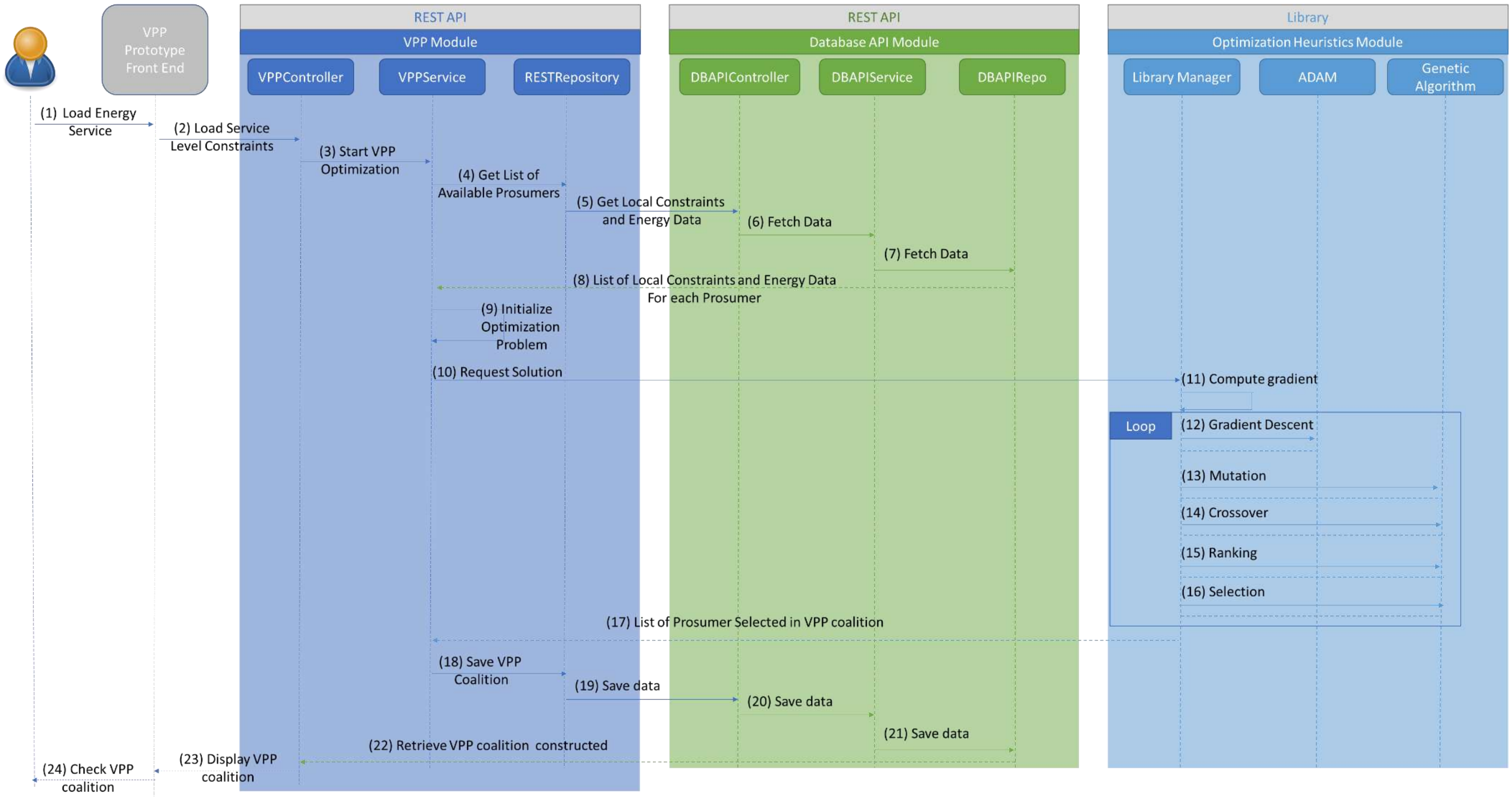


Figure 5. Sequence diagram for VPP prototype modules interaction

Figure 5 presents the sequence diagram for the main modules that are interacting with the goal of creating a new VPP prosumers coalition to address the selected energy service technical constraints. The main steps are the following:

- The VPP manager creates loads a specific energy service request using the VPP Prototype Front End;
- The loaded energy service constraints are forwarded to the VPP module;
- The VPP module will start processing the request, and if no prosumers are specified in the body of the request, it will search for available prosumers for the requested service delivery period;
- The VPP module will retrieve for the set of available prosumers the list of each individual operational constraints as well as their forecasted energy profiles and creates a new VPP Optimization Problem using the model summarized in Section 2.1;
- The VPP module will call the Optimization Heuristics Module to solve the VPP optimization problem and obtain the VPP coalition of prosumers that is able to deliver and meet energy service constraints;
- The Optimization Heuristics Module will perform a setup step in which the gradient for the optimization function is computed, and the parameters for the heuristic's algorithm are set according to the VPP optimization problem at hand;
- The Optimization Heuristics Module will execute a loop with 5 steps on the VPP optimization problem parameters:
    - Compute the values of the real value parameters using a gradient descent-based algorithm;
    - Perform mutation on the integer value parameters by adjusting the value of a parameter slightly;
    - Perform crossover on integer value parameters by interchanging two values from two different solutions;
    - Rank the current solutions by evaluating each solution and rank them according to how close they are to the requested demand;
    - Select the best solutions by trimming the list of solutions based on rank.
- The loop will end when the number of iterations reaches a set limit, or the best VPP coalition of prosumers solution remain unchanged for a significant amount of time;
- When the loop ends, the best result is selected, and the VPP prosumers coalition is returned to the VPP module which saves it in the database;
- Finally, the information on the created VPP prosumers coalition and expected service delivery values are returned to the VPP manager.

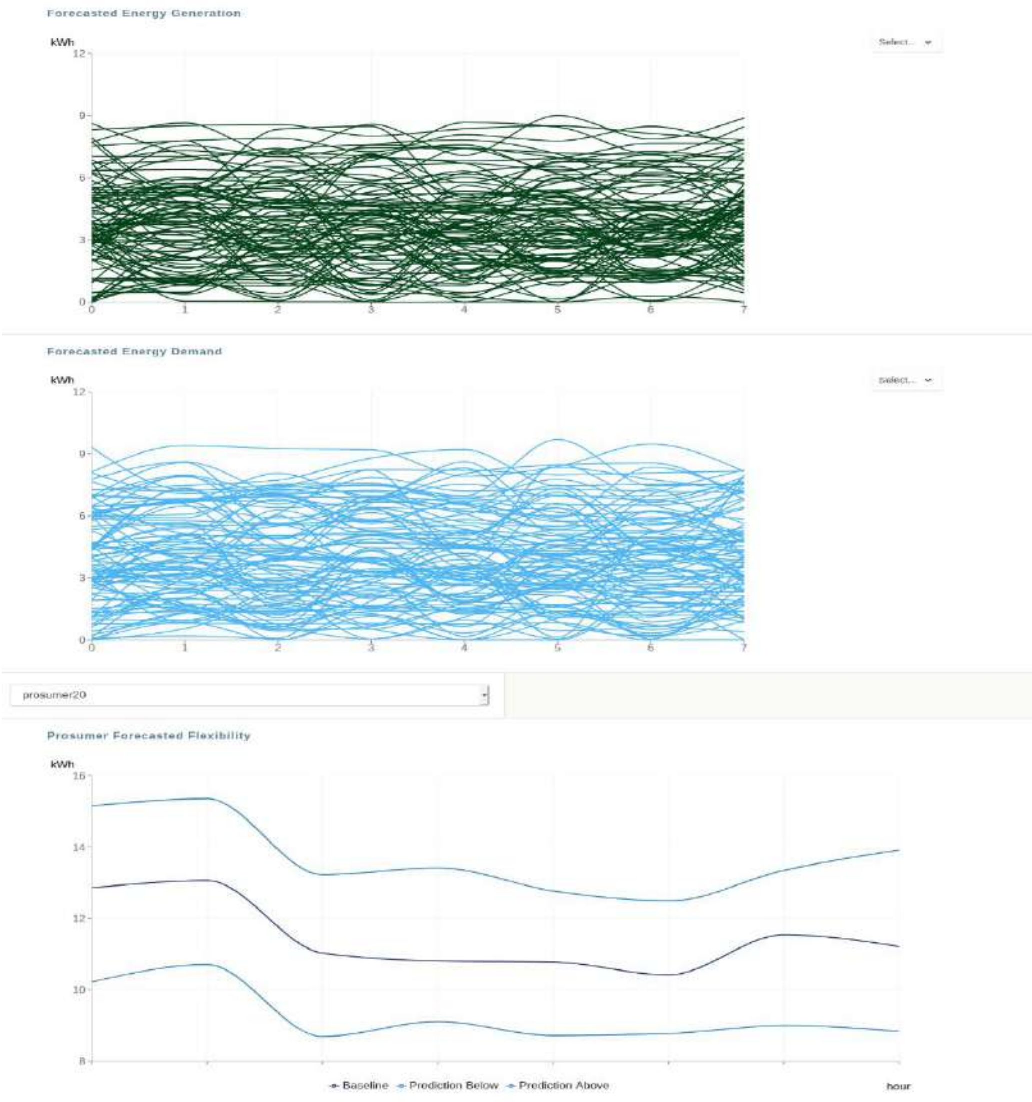
The implemented VPP prototype modules are dockerized [1] for better isolation, the minimum hardware and software requirements being presented in Table 2.

Table 2. VPP prototype requiments

Software requirements	Hardware requirements
64-bit operating system, Java Runtime Environment version 8 or higher, Tomcat Server, MySQL and Cassandra Databases	Processor with at least 4 cores and a base frequency of 2.5GHz, 8 GB RAM, HDD/SSD with 50 GB storage space, Internet connection

The VPP Optimization Module is a Spring Boot project [2] that requires a Tomcat server instance to run. Once deployed, the prototype can be accessed at http://HOST_IP:8083/edream-vpp.

The prototype Front End Web Pages module is implemented using React JS [3] that runs on a web server in a separate Docker container. After deployment it can be accessed at http://HOST_IP:3000/.



hour			
Hour	Baseline	Flexibility Below	Flexibility Above
0	12.85	10.22	15.16
1	13.06	10.70	15.36
2	11.03	8.69	13.23
3	10.81	9.10	13.42
4	10.77	8.71	12.77
5	10.42	8.76	12.48
6	11.54	9.00	13.35
7	11.21	8.83	13.91

Figure 6. Information of prosumers portofolio available for the VPP manager

By navigating on the header bar of the main interface to the “Virtual Power Plants” menu one can access the provided functionalities for VPP dynamic collations formation.

On the first page, the VPP manager can find information about all prosumers willing to join a VPP coalition (see Figure 6). The first two charts display the forecasted energy generation and demand profiles of the prosumers, and the user can toggle each individual prosumer using the dropdown menu. The energy profiles are displayed for a duration of 4 hours, with a granularity of 30 minutes. The elements on the bottom of the page (i.e. a graph and a table) provide information about prosumer forecasted flexibility availability. The user can select the desired prosumer using the dropdown menu located above the graph, and the flexibility information will be loaded. The values represent a lower and upper bound for the production prediction, and they are provided at a granularity of 30 minutes for a period of 4 hours.

On the second web page, the VPP manager can select one of 4 different services that may be targeted by creating a coalition of prosumers from the set of available ones (see Figure 7):

- Trade Energy
- Capacity Bidding/Selling
- Reactive power control
- VPP Demand Response

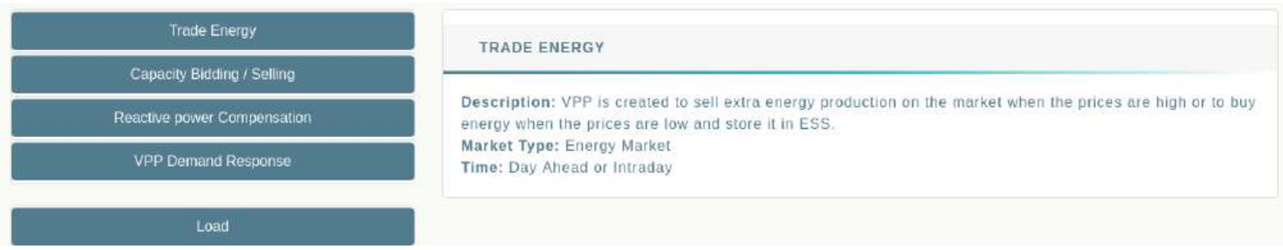
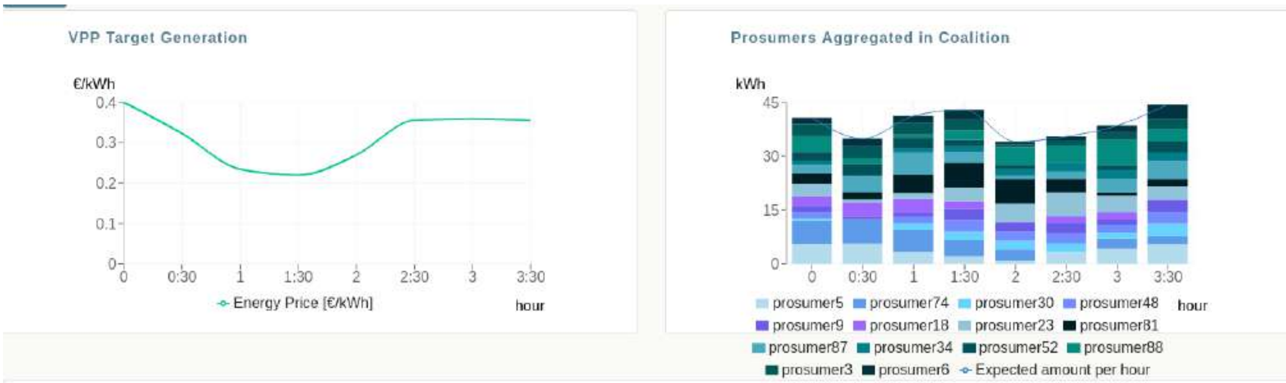


Figure 7. VPP manager interface for selecting a specific service from the list provided

The VPP manager can select by pressing the “Load” button and as a result, a collation will be created to meet the provided service and prosumers constrains.

In case of selecting the *Energy Trading service*, the coalition of prosumers will be created and the following information is displayed (see Figure 8):

- The top left chart shows the service level constraints such as the target energy generation profile and associated price profile.
- The top right chart contains the list of prosumers aggregated in the coalition. Each prosumer is shown as a bar with a different colour which represent the amount of energy it must deliver, and their aggregated total amount represented with a line.
- The table on the bottom contains specific local constraints and target objectives of each of the prosumers selected in the coalition.



hour		prosumer		
Hour	Prosumer	Request	Flexibility Below	Flexibility Above
0	prosumer5	7.00	5.65	8.43
1	prosumer5	7.00	5.39	8.30
2	prosumer5	8.43	6.82	10.14
3	prosumer5	8.78	6.79	10.39
4	prosumer5	8.26	6.41	9.87
5	prosumer5	8.28	6.58	9.81
6	prosumer5	6.27	4.94	7.42
7	prosumer5	6.25	4.97	7.39

Previous Page 1 of 12 10 rows Next

Figure 8. Interface showing information about the VPP coalition of prosumers created to trade energy

In case of selecting the *Capacity Bidding / Selling service* the coalition of prosumers will be created and the following information is displayed (see Figure 9):

- The top left chart shows the service level constraints such as the target capacity needed to be delivered.
- The top right chart contains the list of prosumers aggregated in the coalition. Each prosumer is shown as a bar with a different colour which represent the amount of energy it must deliver, and their aggregated total amount represented with a line.
- The table on the bottom contains specific local constraints and target objectives of each of the prosumers selected in the coalition.

hour			prosumer	
Hour	Prosumer	Request	Flexibility Below	Flexibility Above
0	prosumer7	9.05	7.27	10.68
1	prosumer7	9.12	7.62	11.19
2	prosumer7	10.32	8.23	12.19
3	prosumer7	10.33	7.98	12.44
4	prosumer7	8.58	6.69	10.46
5	prosumer7	8.79	6.91	10.36
6	prosumer7	8.99	7.07	10.45
7	prosumer7	8.82	7.22	10.47

Previous Page 1 of 7 10 rows Next

Figure 9. Web page showing information about the VPP coalition of prosumers created for capacity selling

In case of selecting the *Reactive Power Control service* the coalition of prosumers will be created and the information is displayed on two web pages. The first web page contains information about the grid power factor before and after deployment of the VPP (see Figure 10).

Figure 10. Web page showing information about grid power factor before and after VPP coalition deployment

In the second web page, the following information about the VPP coalition of prosumers created to provide reactive control service is provided (see Figure 11):

- On the top left and right charts, the active and reactive power values for the coalition, respectively;
- In the middle chart the coalition power factor;
- The table on the bottom contains specific local constraints and target objectives of each of the prosumers selected in the coalition.

hour | prosumer

Hour	Prosumer	Request	Flexibility Below	Flexibility Above
0	prosumer85	9.65	7.52	11.38
1	prosumer85	9.33	7.47	11.57
2	prosumer85	10.90	8.96	13.72
3	prosumer85	11.45	9.10	13.26
4	prosumer85	11.30	9.19	13.61
5	prosumer85	11.81	9.50	13.56
6	prosumer85	8.64	6.57	10.30
7	prosumer85	8.47	6.82	9.97

Figure 11. Web page showing information about the VPP coalition of prosumers created delivery of reactive power control

In case of selecting the *DR service* the coalition of prosumers will be created and the following information is displayed on the web page (see Figure 12):

- The top left chart shows the service level constraints such as the demanded profile to be delivered.
- The top right chart contains the list of prosumers aggregated in the coalition. Each prosumer is shown as a bar with a different colour which represent the amount of energy it must deliver, and their aggregated total amount represented with a line.
- The table on the bottom contains specific local constraints and target objectives of each of the prosumers selected in the coalition.

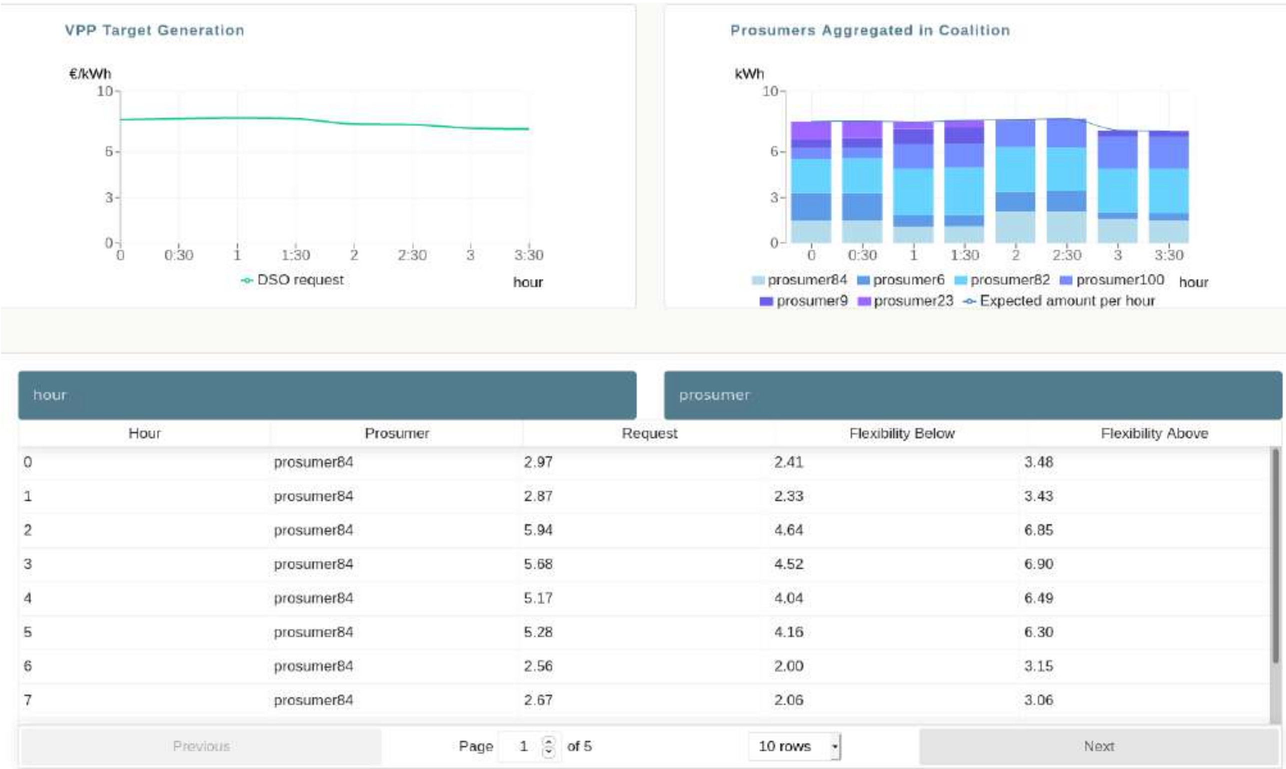


Figure 12. Web page showing information about the VPP coalition of prosumers created for addressing a DR service

2.3 Evaluation results

In this section we present evaluation results regarding the capability of our VPP construction and optimization model to solve at the fog level, the VPP specific constraints satisfaction problems and by generating the prosumers coalitions at different levels in the hierarchy tailored to technical constraints of reactive power control service.

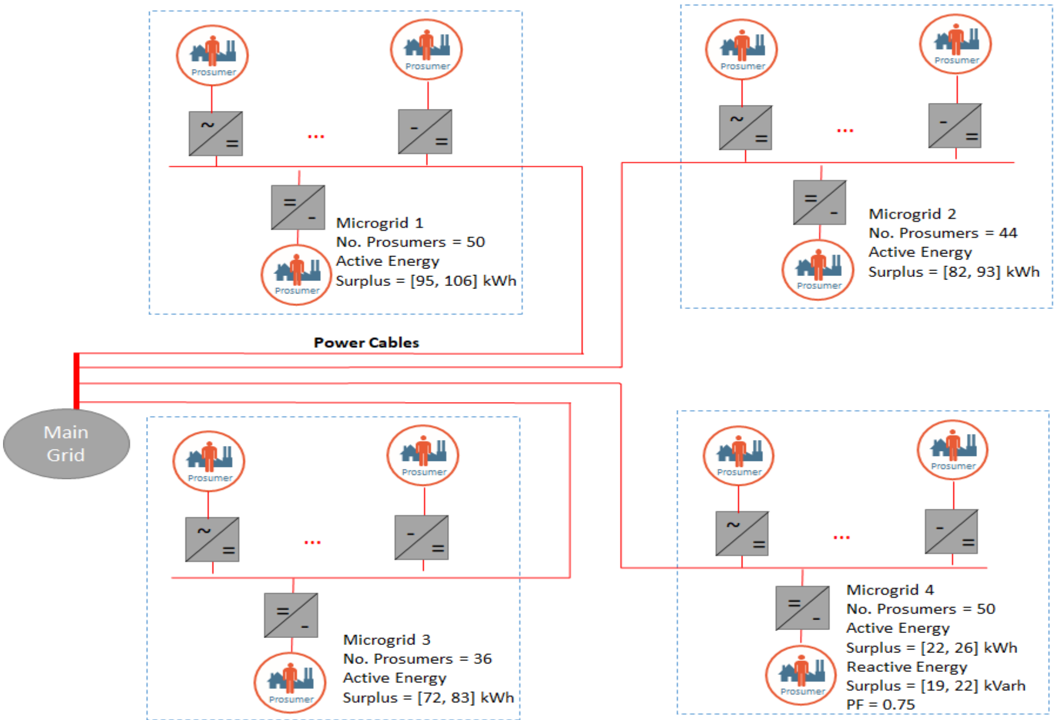


Figure 13. Scenario considered for evaluation

We have defined an evaluation scenario (see Figure 13) in which we have simulated four micro-grids (ids 1 to 4) with a various number of prosumers. The first three of them present a potential surplus of active energy by activating of additional generation prosumers, whereas the fourth one presents a low PF value due to an actual surplus of reactive energy (i.e. micro-grid 4). Table 3 presents the ranges of parameters in which the prosumers part of each microgrid were randomly generated. Detailed description of the simulation setup can be found in our paper published here [4].

Table 3. Experimental setup for prosumers generation

Experimental Setup	Min	Max
Prosumer Generation	0	3.34 kWh
Forecasting model uncertainty	0%	10%

The VPP model and optimization will address the low power factor in microgrid 4 by constructing a VPP with the aim of virtually aggregating the surplus of active power available at microgrids 1-3 to balance and improve the low power factor from microgrid 4 (i.e. from the low level of 0.75 and close to 0.9).

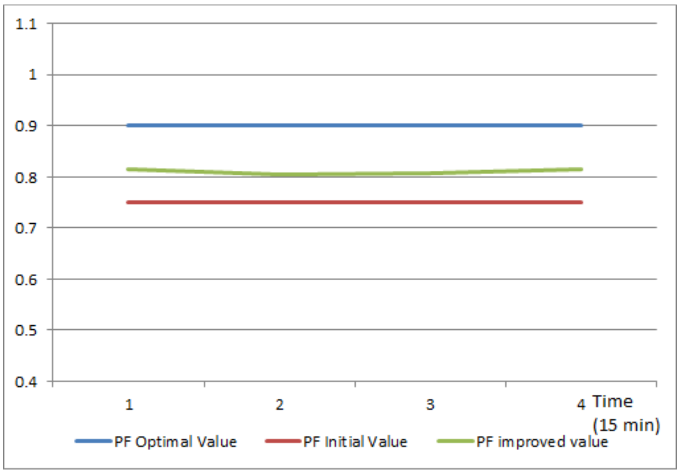


Figure 14. Power factor improvement as result of VPP construction and service delivery

Figure 14 shows the power factor value in the initial situation (the red line), the optimal value targeted in the frequency restoration reserve service (the blue line) and the PF value after the new second level VPP is created, activated and used to compensate the reactive power from microgrid 4 (the green line). As a result, the PF value gets close to the optimal value stabilizing the voltage thus the potential unbalances are locally addressed and not escalated to the main grid level.

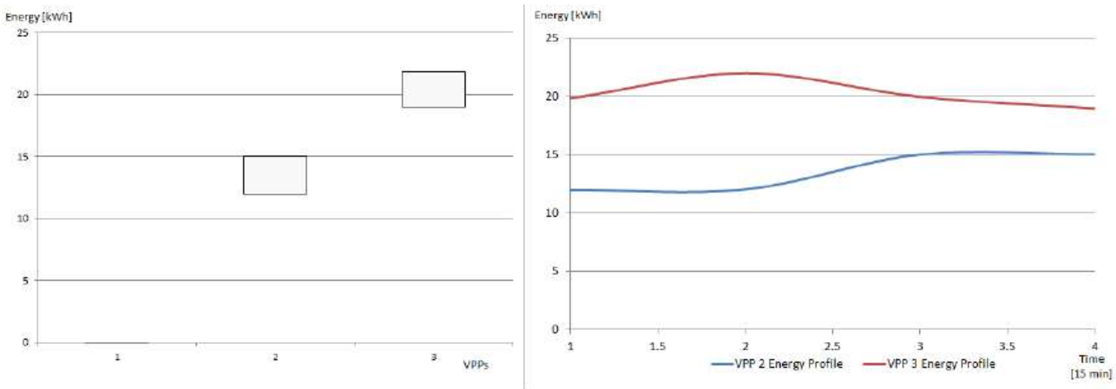


Figure 15. New created hierarchical VPPs: (LEFT) maximum and minimum energy consumption in the service interval and (RIGHT) detailed energy profiles

The active power profiles of level one VPP2 and VPP3 corresponding to the microgrids 2 and 3 (see Figure 15) consisting of 16 prosumers activated from the microgrid micro-grids to create the higher level to compensate for the low PF of microgrid 4. Each prosumer has a different PF value, being able to compensate together for the imbalance between the active and reactive power from the grid. The final grid situation is shown in Figure 16. The reactive power has been decreased by adding additional energy prosumers with a leading power factor. Furthermore, the active power in the grid has increased with around 30 kW, because the new prosumers have been activated and grouped in the VPP, whose total active power contribution is shown with the red.

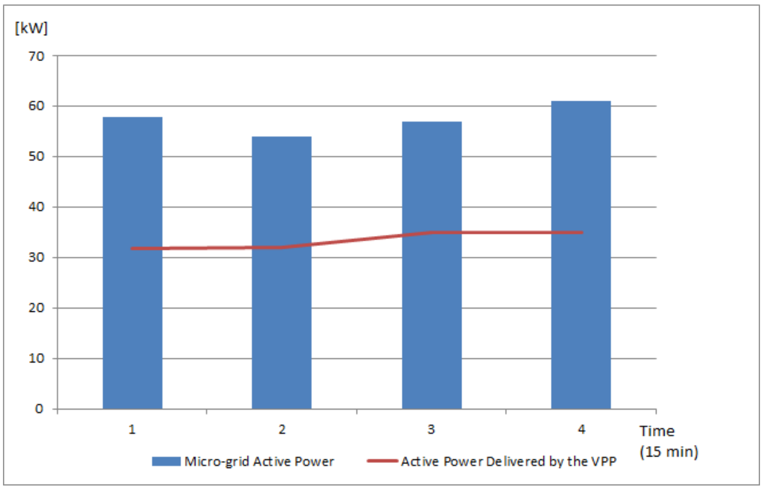


Figure 16. Micro-grid total active power and VPP contribution

Finally, we have determined the benefits of using a fog-edge based architecture for data collection and virtual power plant dynamic coalition creation considering the number of messages sent to the prosumers for services activation. We have assumed that we have a monitoring system based on smart meters deployed at the prosumer side that have a sampling rate of 5 seconds. In the classical architecture, all the data is collected to a central point (i.e. in the cloud), while in our fog-edge architecture, data is stored at microgrid level. At the same time, the aFRR activation message is now sent only once from the cloud level to the fog level, where the microgrid associated VPPs are being created and only then distributed to all newly identified prosumers for their activation. Figure 17 shows the significant decrease in the number of messages used in case of a decentralized control at fog level compared the number of data messages sent over the communication network in a cloud-based architecture.

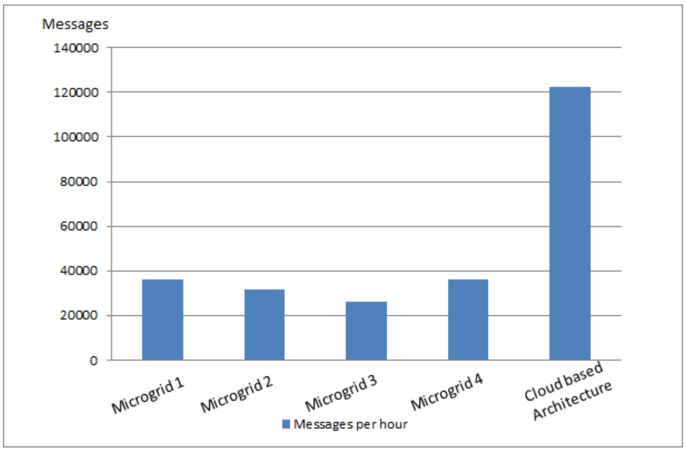


Figure 17. Number of messages send during one hour by smart meters associated with prosumers

3 Blockchain based VPP

In this section we present a mechanism for on-the-fly construction of VPP coalitions of prosumers in a decentralized fashion leveraging on a blockchain based infrastructure.

3.1 Smart contracts for decentralized coalition formation

The blockchain based mechanism for prosumers coalitions formation in VPPs will provide trackable, auditable and decentralized management features enabling small producers to increase their profit by aggregating and delivering higher quantities of energy at once meeting the entry cap limits on different markets. The mechanism enforces each prosumer responsibility in the delivery of the promised energy values by increasing or decreasing the risk/reward ratio based on their own actions, while at the same time considering the uncertainty of the energy forecasting process.

Blockchain systems can guarantee decentralization, although, due to security reasons, each instruction executed on blockchain has an additional overhead. For a solution to be feasible on blockchain, it must be very lightweight in terms of complexity and time. Because most solutions that optimally construct coalitions of prosumers in VPPs use very high computation algorithms, they are not usually suited for a full implementation on blockchain, thus alternative solutions based on Oracles based integration of external services may be adopted.

Anyway, our proposed blockchain mechanism innovatively addresses the VPP coalition construction using a lightweight algorithm implemented using smart contracts that targets the construction of hierarchical VPP structures enabling its efficient running completely on blockchain (i.e. no Oracles are used). The hierarchical VPP has a tree like structure, with a root VPP which is the initiator of the coalition and lower scale VPPs and all intermediary levels while the prosumers will be on the leaf.

We have modelled the Grid as a weigthed graph of energy prosumers $G(V, E)$, where the vertices *V* are the energy prosumers and the edges *E* represent the transmission lines between them.

$$Smart_{Grid} = \{G(V,E) | V = \{Prosumer_k | k = 1..N\}, E = \{e_{ij} | i,j = 1..N\}\}$$

In terms of connectivity, *G* is a complete graph, because each pair of prosumers can be connected in order to deliver or receive energy. From this graph, the VPP construction mechanism will select an acyclic subgraph, forming a tree, in which each child will deliver specific energy service to its parent, and each parent will pay its children according to the actual monitored service delivery. A single node cannot be contained in more than one tree at a given time.

This interaction is managed using a VPP construction smart contract, which can form a VPP hierarchical structure, becoming the root or internal node of that network, or join as simple prosumer in an already created one, becoming a leaf in the network. Figure 18 describes the main interaction between a child VPP and a parent VPP as it is regulated by the state changes in smart contracts execution. To join a VPP coalition, a prosumer must receive a join service offer which contains the services level constraints that it must deliver and the associated financial payment for success delivery. After the delivering timeframe, based on the actual monitored delivery values the parent VPP will do the financial settlement, i.e. the method that is recursively called for every child.

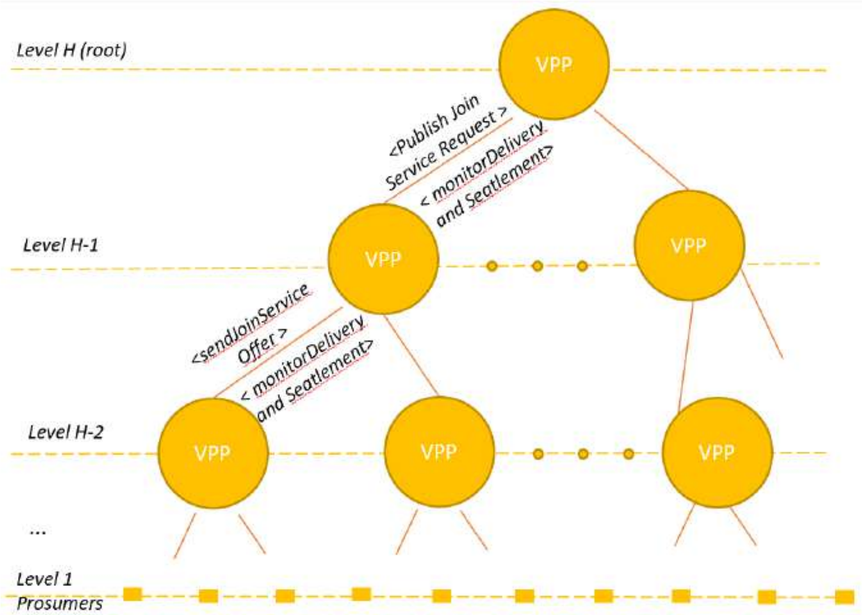


Figure 18. Interaction between smart contracts in a hierachical VPP structure

The advantage of our proposed approach is that it is lightweight and does not need high computation time, so it is feasible to run on blockchain, without third party software outside the blockchain, like Oracles. The algorithm can be implemented using a linear iterative insert, like the insertion sort algorithm, or using a Balanced Binary Search Tree, reducing the linear complexity to $O(\log n)$ for each insert operation. The linear insert is favoured in case of small number of members in a coalition, and because it does not involve recursion, a very expensive feature on blockchain. As a result the descentralized mechansim for VPP coaltions creation can be mapped to blockchain techology by leveraging on self enforcing smart contracts.

The interaction among smart contracts involved in a hierarchical VPP structure is depicted in Figure 19.

- Step 3.1: A prosumer sees that there is a need for additional capacity, so it tries to aggregate that capacity by building the hierarchical structure of VPP coalitions of prosumers. Using the smart contract regulating its behaviour, it publishes a join service request specifying the price per energy unit, the quantity and the timeframe when the energy should be delivered. As a result, a joint request session is created, and the prosumer will become the initiator of VPP coalition formation and the root of the VPP hierarchical structure.
- Step 3.2: Based on the current service request, all notified prosumers can compete for a place in the VPP hierarchy, either as individual prosumers, or as new lower level VPP coalitions. In both cases, the prosumer will offer a price per energy unit for a specific percentage of the requested amount of energy it may deliver. If a prosumer decides to join as a VPP coalition, the process repeats recursively, making this prosumer an internal node of the network. If it opts to join individually, the prosumer becomes a leaf in the network. The only requirement before joining a VPP is to deposit the tokens associated with the amount of energy promised to be delivered to the parent VPP. This feature increases security in case the prosumer will not be able to deliver the whole quantity some of the deposit tokens will be lost.
- Step 3.3: Each offer made to a higher VPP level will cause that VPP to rebalance and improve its coalition. The algorithm keeps offers sorted in ascending order by price per unit of energy. Each time a new prosumer joins the coalition, with a price lower than the already most expensive offer, the

quantities of each prosumer will be shifted in a way that will keep the initial values of the total amount of energy the VPP coalition had aggregated.

- Step 3.4: Right before the energy delivering timeframe starts, the final states of each VPP collation of prosumer from the hierarchy is saved. The prosumers that had their offers entirely rejected by their parent VPP will get the deposits back, while the prosumers with offers only partially matched will get a percentage of the deposit, equal with the percentage of the quantity of energy not matched.

- Step 3.5: During the service delivery timeframe, each prosumer smart contract will track and register the monitored energy values (i.e. that actual amount of energy produced and delivered). The energy monitored values will be registered by the end of delivery session, at which point the values will be used by the parent VPP for the financial settlement of the services.

- Step 3.6: The root VPP will stop the VPP session after the timeframe, by calling the evaluate function for each of his children and paying the price for the energy. Recursively each lower level VPP will evaluate the delivered energy against the promised energy for each of his children. In case, of an individual prosumer child, the recursive hierarchical structure construction process will stop. After the monitoring of energy delivery, each parent VPP will conduct the settlement of the service both from energy and financial perspectives.

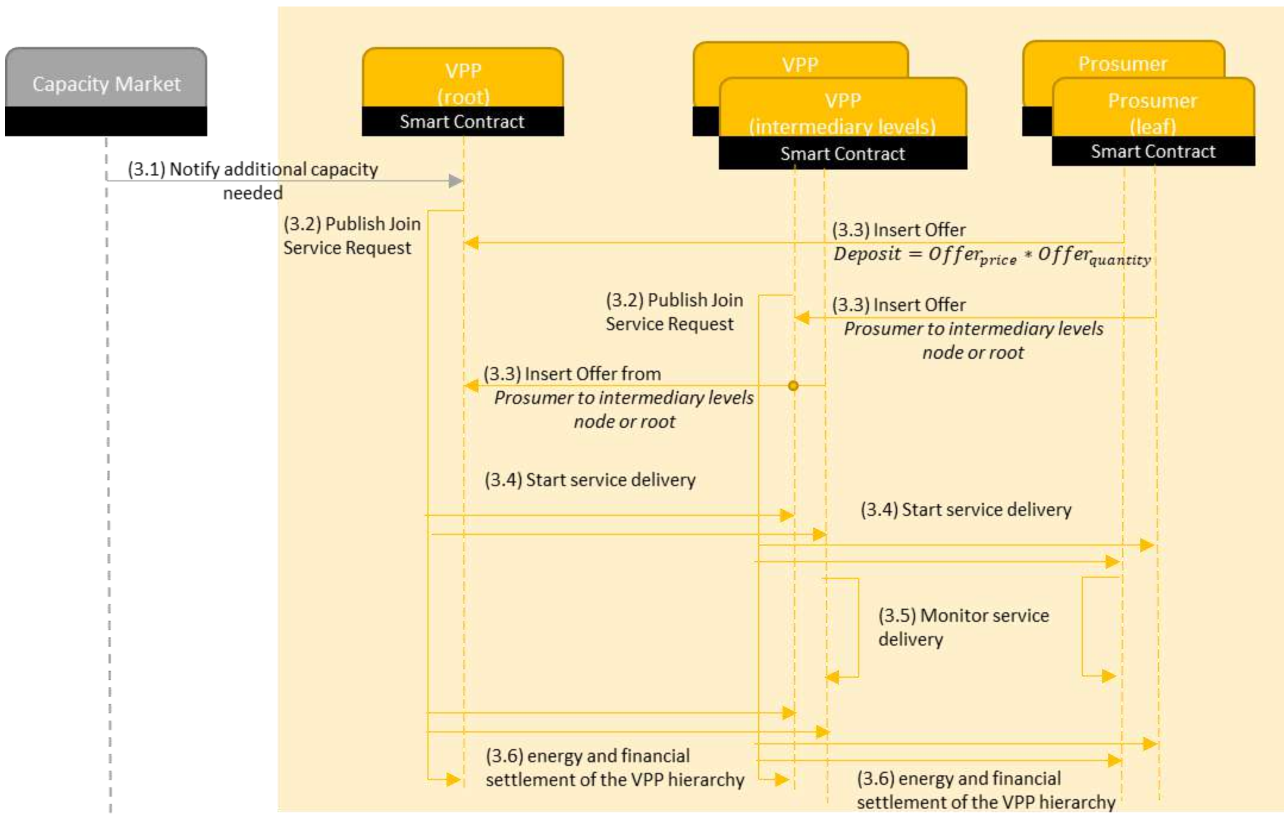


Figure 19. Sequence diagram for decentralized VPP coalition formation using smart contracts

The algorithm used to form a coalition is an online and decentralized one that re-evaluates the state of the coalition, each time a new offer for joining the VPP is registered. Any time, while the VPP is still accepting collation joint offers, the member is sorted in ascending order by their price per unit of energy, independent of the quantity of the offer, from left to right (cheapest price is the leftmost offer). When a new member joins the new offer, he is inserted in the set and the algorithm will place the order in the right position, by shifting the more expensive offers to the right, by applying an insert operation similarly to the insertion sort algorithm. When the session finishes, the algorithm will return the first offers (from left to right) that can deliver the total amount of energy expected by the VPP.

The prosumer that is initializing a new hierarchical VPP construction session, will publish a service request, where other prosumers can offer to join to deliver energy. In this scenario, the prosumer will specify, the start time and end time of the delivery period, price per unit and total amount of energy needed (see Figure 20). In the smart contract managing the prosumers behaviour this functionality is achieved using a function which is called each time a prosumer wants to open a VPP creation session, either as root or as internal node.

```
1  function sendOffer(uint price, uint quantity, address payable vppManagerAddress) public payable{
2      require(_ownerVpp == msg.sender);
3      require(msg.value >= price * quantity);
4  
5      /*Call VppManager*/
6      VppMember vppManager = VppMember(vppManagerAddress);
7      vppManager.receiveOfferManager.value(msg.value)(price, quantity, _risk);
8      _vppManagerAddress = vppManagerAddress;
9  
10     emit VppOfferSent(address(this), vppManagerAddress, msg.value, getNowDate());
11     emit VppPersonalInfo(_risk, _vppManagerAddress, _monitoredValues, getNowDate());
12 }
```

Figure 20. Smart contract method for publish a service request offer

Suppose one prosumer wants to create at level $h + 1$ a VPP coalition aggregating lower level VPPs (i.e. from level h). The child prosumer/VPP will call his own contract method*,* depicted in Figure 20 using the price per unit, quantity and the address of the of the parent associated smart contract, as parameters. The transaction signature is validated (see line 2) in order to ensure that only the transactions signed with the private key owned by the contract's owner are considered, thus preventing any malicious activity. Furthermore, the prosumer is required to provide an assurance deposit (see line 3) for the promised energy to be delivered, in order to ensure responsible behaviour during the delivering period. Upon validation, the function will call (see line 7) the parent contract *receiveOfferManager* function*,* that is responsible to register the prosumer's offer and run the rebalance algorithm (see Figure 21) in order to form the VPP coalition.

```
1  function receiveOfferManager(uint price, uint quantity, uint risk) public payable{
2      require(_vppGrid.active == 1);
3      require((now <= _vppGrid.endTime) && (_vppGrid.startTime <= now));
4      require(msg.value >= price * quantity);
5      _vppGrid.memberAddress.push(msg.sender);
6      _vppGrid.memberDeposit.push(msg.value);
7      _vppGrid.memberPromisedQuantity.push(0);
8      if(_ownerVpp == tx.origin){
9          _vppGrid.memberRisk.push(0);
10         _vppGrid.memberPrice.push(0);
11         _vppGrid.memberQuantity.push(quantity);
12     }else{
13         _vppGrid.memberRisk.push(risk);
14         _vppGrid.memberPrice.push(price);
15         _vppGrid.memberQuantity.push(quantity);
16     }
17     for(int256 i = int256(_vppGrid.memberAddress.length) - 2; i >= 0; --i){
18         uint256 ind = uint256(i);
19         if(_vppGrid.memberPrice[ind] * _vppGrid.memberRisk[ind] >
20             _vppGrid.memberPrice[ind + 1] * _vppGrid.memberRisk[ind + 1]){...} //Swap
40     }
41     emit VppGridChanged(_vppGrid, getNowDate());
42 }
```

Figure 21. Smart contract method for receiving a join VPP coalition offer

As observed, the reordering is based on an insert method (see lines 17-40 in Figure 21), that orders each member of the coalition in ascending order. Due to its simplicity and low complexity, this algorithm works fine even if it runs entirely on blockchain. This method also registers the tokens deposits for each prosumer (line 6 in Figure 21) that wants to participate in the VPP coalition formation.

$$\sum memberDeposit_k = \sum price_i * quantity_i,\ \forall\, k \in V, i\ \in V \mid parent_i = k$$

Before the actual energy delivery session, the root of the hierarchical VPP structure will evaluate and finalize the construction session. In this step, the coalition is determined by taking the first prosumers that sum to the total quantity. As previously described, each contract has its own security validation (see line 2 in Figure 22). The last prosumer's quantity will be split, if the total sum is greater than the energy requested by the VPP manager (line 18 in Figure 22).

```
1  function evaluateAndStopMarketManager() public {
2      require(_ownerVpp == msg.sender);
3      require(_vppGrid.active == 1);
4      require(now > _vppGrid.endTime);
5      _vppGrid.active = 2;
6      uint currentQuantity = 0;
7      int lastIndex = -1;
8      uint256 lastSurplus = 0;
9      for(uint256 i = 0; i < _vppGrid.memberAddress.length; ++i){
10         if(currentQuantity + _vppGrid.memberQuantity[i] >= _vppGrid.quantity){
11             lastIndex = int(i);
12             lastSurplus = currentQuantity + _vppGrid.memberQuantity[i] - _vppGrid.quantity;
13             break;
14         }else{
15             currentQuantity += _vppGrid.memberQuantity[i];
16         }
17     }
18     if(lastIndex == -1){…}else{…}   //Split offer in case of surplus
30     emit VppGridChanged(_vppGrid, getNowDate());
31 }
```

Figure 22. Smart contract method for finishing the VPP hierarchy construction process

The final step concerns the tracking and assessing of the actual delivery of energy and conducting the financial settlement of prosumers accounts (see Figure 23). This method is called only by the root of the hierarchy, because it will recursively do the settlement on all levels. Consider a prosumer acting as a VPP manager at height $h-1$ in the hierarchy. His parent contract from height h will announce by calling the settlement function that the VPP is conducting the energy and financial delivery check. The contract on level h will announce recursively (lines 4-31 in Figure 23) all his children contracts on the $h-2$ level, and so on. After the recursive calls terminate, the state of the VPP hierarchy will be sealed (lines 32-43 see Figure 23).

```
1  function payAndUpdateRiskManager() public{
2      require((_ownerVpp == msg.sender && _vppManagerAddress == address(0)) || _vppManagerAddress == msg.sender);
3      require(_vppGrid.active == 2 || _vppGrid.active == 0);
4      for(uint256 i = 0; i < _vppGrid.memberAddress.length; ++i){…} //Recursively call payAndUpdateRiskManager
31                                                                    //for each child; pay and update each child.
32     _vppGrid.active = 0;
33     _vppGrid.startTime = 0;
34     _vppGrid.endTime = 0;
35     _vppGrid.price = 0;
36     _vppGrid.quantity = 0;
37     delete _vppManagerAddress;
38     delete _vppGrid.memberAddress;
39     delete _vppGrid.memberRisk;
40     delete _vppGrid.memberPrice;
41     delete _vppGrid.memberQuantity;
42     delete _vppGrid.memberDeposit;
43     delete _vppGrid.memberPromisedQuantity;
44     emit VppGridChanged(_vppGrid, getNowDate());
45 }
```

Figure 23. Smart contract method for energy and financial settlement of the delivery

3.2 Prototype implementation and results

The above presented solution was implemented in a blockchain protype with the architecture presented in Figure 24.

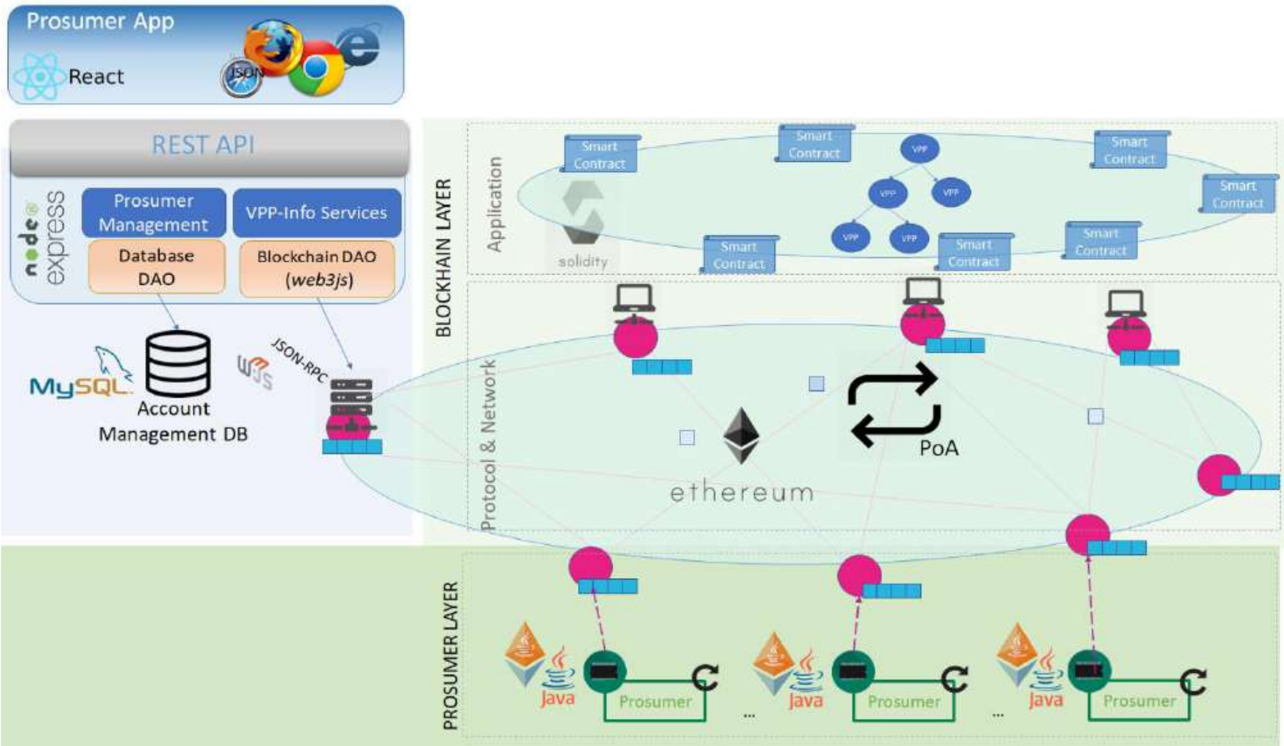


Figure 24. Blockchain based VPP architecture

The **Blockchain Network Layer** is based on a custom Ethereum blockchain network where the smart contracts are deployed, managing the VPP construction for meeting the constraints of the energy services to be delivered. The smart contracts are implemented using Solidity, while Proof of Authority is used as consensus algorithm. We have considered the default mining time of 15 seconds and a gas limit comparable with the public networks one of 8 000 000 gas per block.

The **Application Layer** contains a:

- *Server Application* exposing REST API (Blockchain API) acts as a proxy between the prosumer's client applications and the blockchain platform. This architecture has been chosen to provide fast prototyping features, allowing the simulation of many prosumers, as opposed to fully decentralized client applications that would require deploying several client applications proportional with the number of prosumers registered. The main components of the server-side application are the account management storage, dealing with the registered prosumer accounts and a REST API is implemented using NodeJS, and web3js exposing an API over the database and over the blockchain platform (i.e. manages the smart contracts method calls).
- *React Client Application*, a web application that allows the prosumers to interact with the prototype and visualize the progress in energy services delivery and balance tracking.

The **Edge Prosumer Layer** deals with the energy data monitoring, acquisition and registration the blockchain, by connecting to the Ethereum Node (local or remote) for periodically storing of the values on-chain. For simulation purposes, the data feed is currently simulated based on data set files.

We have created and deployed a more complex smart contract associated with the prosumers, instead of more contracts with less responsibilities, because each prosumer can be a potential VPP initiator/manager or leaf in the hierarchical VPP structure. Combined with the recursive approach, this was the most advantageous way to model the prosumers management alternatives. As a result, we can maintain a hierarchical VPP structure with each node having similar contract for management.

To evaluate the potential of the blockchain platform for creating on-the-fly a hierarchical VPP structure to deliver a capacity service we have considered set of 7 prosumers. All prosumer nodes energy measuring devices were simulated by feeding monitored values every hour, according to the energy data collected from ASM Terni pilot. We considered in our scenario that in the Capacity Market there is a need to cover 120 KWh additional capacity in the time interval between hours 16:00 and 17:00. Prosumer 6 will initialize the hierarchical VPP construction to provide the missing capacity, while the other prosumers will join, as individuals or as part of smaller lower level VPPs.

Figure 25 shows the VPP hierarchical structure created in our simulated scenario to provide the additional capacity needed. VPP6 is the root, becoming the one who publish the service request setting the timeframe for delivering, expected amount of energy needed.

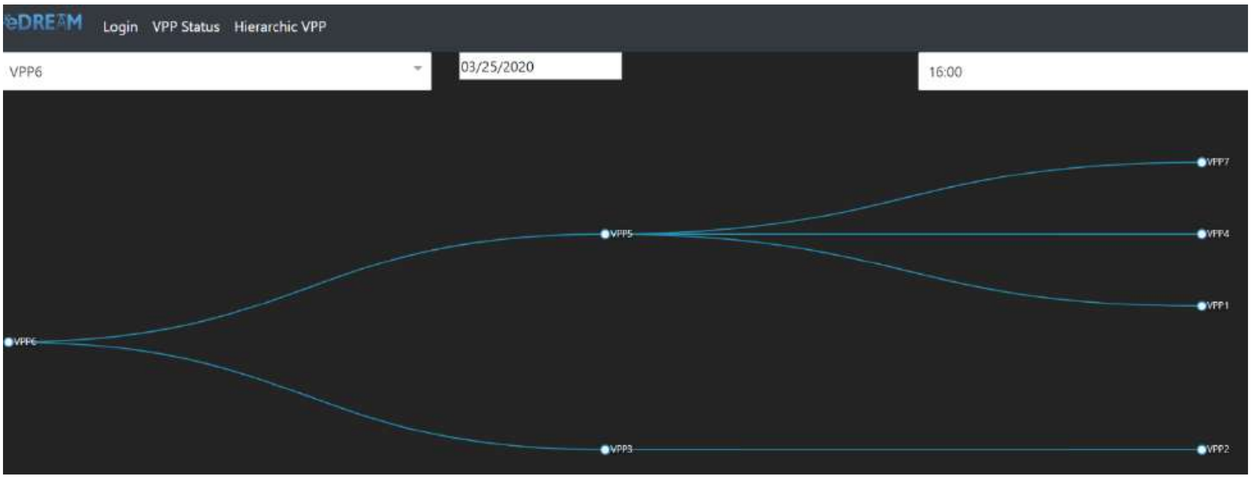


Figure 25. Hierarchical VPP structure created to deliver energy

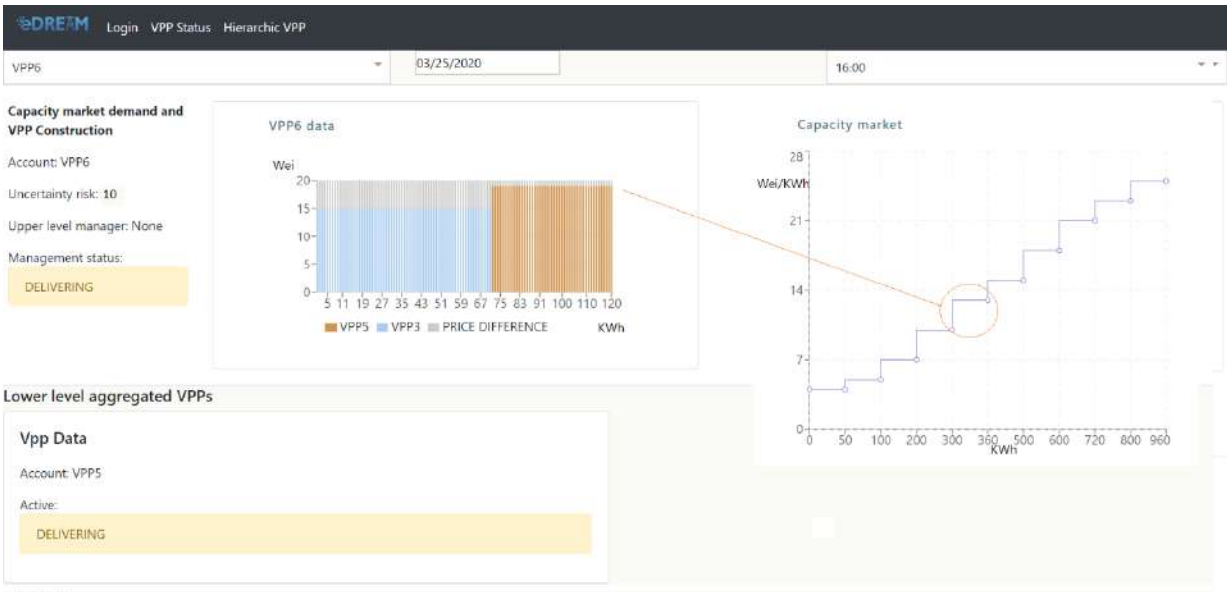


Figure 26. VPP6 root state and aggregated lower level VPPs

The uncertainty risk is initialized to 10 which is the average uncertainty on the forecasting process because the prosumer did not participate before in any coalition. As shown in Figure 26. VPP6 root state and aggregated lower level VPPs VPP6 will group lower level VPPs 3 and 5 in a coalition. At their turn VPP5 (Figure 27) and VPP3 (Figure 28) will activate their lower level VPPs or prosumers to aggregate the amount of energy needed to be delivered at higher level.

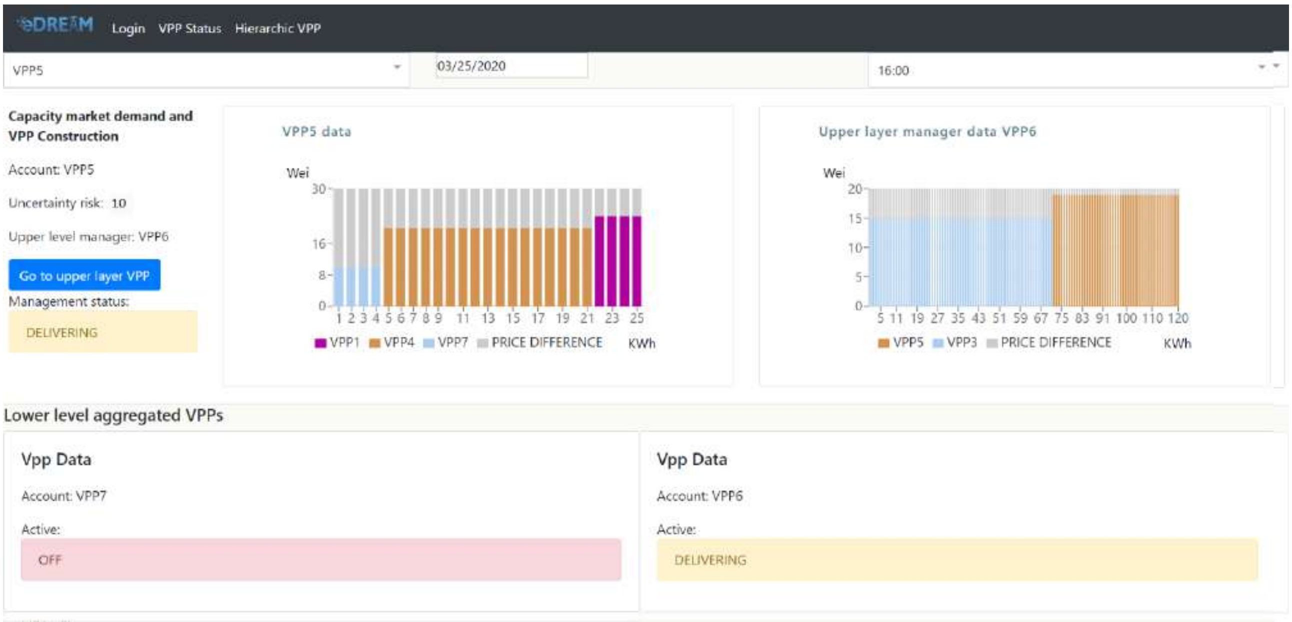


Figure 27. VPP5 state on level 1 and aggregated lower level VPPs/prosumers

Each prosumer can see and track the actual delivery of energy as their associated smart contracts are registering the monitored energy generation values.

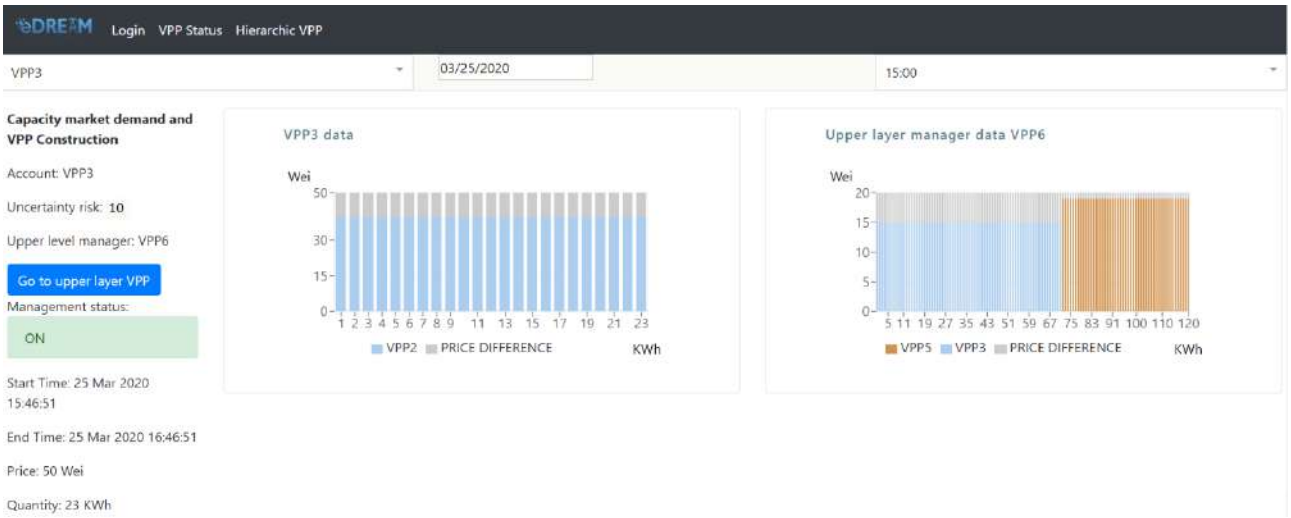


Figure 28. VPP3 state on level 1 and aggregated lower level VPPs/prosumers

After the capacity delivery session is finished, the energy and financial settlement is conducted on the whole hierarchy and each risk is updated and the prosumer nodes are paid, according to the energy delivered.

4 Conclusion

In this deliverable, we had described two different solutions for constructing in a decentralized manner VPP coalitions of prosumers to deliver energy services such as capacity bidding/selling, reactive power control, and VPP demand response.

The first solution is based on a fog-based architecture which enables the on-the-fly consideration of both prosumers constrains and service level constraints for the construction of the VPP coalition. The VPP model and optimization heuristics are deployed at the fog level and are used to solve the VPP construction specific constraints satisfaction problems. The results show that the defined fog enabled solutions can construct VPPs coalitions with low computational overhead (i.e. number of messages exchanged).

The second solution uses the blockchain technology for the construction of VPP coalitions of prosumers by integrating a lightweight mechanism that is implemented leveraging on self-smart contracts as well as VPP service delivery and settlement process. The evaluation results show that it can be used for managing the VPP delivery of capacity service with low computational time making it appropriate to run on the blockchain, without third-party software outside the blockchain, such as Oracles.

References

[1] Docker, https://www.docker.com/
[2] Spring Framework. [Online]. Available: https://spring.io/
[3] React framework. [Online]. Available: https://reactjs.org/
[4] Pop, C.; Antal, M.; Cioara, T.; Anghel, I.; Salomie, I.; Bertoncini, M. A Fog Computing Enabled Virtual Power Plant Model for Delivery of Frequency Restoration Reserve Services. Sensors 2019, 19, 4688.
[5] EDREAM deliverable D3.3 "Consumption flexibility models and aggregation techniques"